Esta monografia foi julgada adequada como **TRABALHO DE CONCLUSÃO DE CURSO** no curso de Matemática - Habilitação Bacharelado e aprovada em sua forma final pela Banca Examinadora designada pela Portaria nº. 78 / SGC / 04.

Profª. Carmen Suzane Comitre Gimenez

Professora responsável pela disciplina

Banca examinadora:

Mário César Zambaldi

Orientador

Daniel Norberto Kozakevich

Márcio Rodolfo Fernandes

GRASIELLI GAVA

Resolução de Problemas de Minimização com Restrições Lineares de Igualdade

Trabalho de Conclusão de Curso apresentado ao
Curso de Matemática - Habilitação Bacharelado
Departamento de Matemática
Centro de Ciências Físicas e Matemáticas
Universidade Federal de Santa Catarina

Orientador: Mário César Zambaldi

Florianópolis
Dezembro 2004

Sumário

Introdução

Modelos matemáticos em diversas áreas das ciências aplicadas originam problemas de otimização que devem ser abordados por métodos numéricos. Na medida em que os modelos se tornam mais complicados, novas metodologias devem ser desenvolvidas. Os métodos para resolução de problemas de otimização são baseados, em geral, em teorias consistentes e algoritmos computacionais robustos.

Neste trabalho consideramos um problema típico e importante de otimização: a resolução de modelos matemáticos com minimização de uma função linear com restrições de igualdades lineares. Muitos modelos importantes apresentam esta formulação. Para implementação computacional e validação dos testes numéricos empregamos o ambiente CUTE (Constrained and Unconstrained Testing Environment), um ambiente robusto de otimização numérica em que estão disponíveis vários modelos para avaliação de métodos numéricos. O principal objetivo é a compreensão dos métodos apresentados, assim como a utilização de uma poderosa plataforma para desenvolvimento e testes de algoritmos em otimização.

O trabalho está organizado da seguinte maneira. No capítulo 1 falamos um pouco sobre a minimização irrestrita e sobre alguns métodos numéricos para resolver o problema de minimização irrestrita , no capítulo 2 desenvolvemos toda a teoria de minimização com restrições lineares de igualdade, ressaltando o efeito das restrições lineares de igualdade no tamanho do problema. O capítulo 3 mostra a relação da fatoração QR com os quatro espaços fundamentais e no capítulo 4 são apresentados os algoritmos e os testes numéricos. Finalmente apresentamos as conclusões e futuras possibilidades de trabalhos.

1 Minimização Irrestrita

Um importante problema de otimização é a minimização de uma função objetivo de variáveis reais, sem nenhuma restrição nessas variáveis. Em termos matemáticos, temos:

$$\underset{x \in \mathbb{R}^n}{\text{minimizar}} \, f(x) \tag{1}$$

onde $x \in \mathbb{R}^n$ e $f : \mathbb{R}^n \longrightarrow \mathbb{R}$ é uma função regular. Vamos então expor, como reconhecer as soluções:

1.1 A Solução de um Problema de Minimização Irrestrita

Um minimizador de f é um ponto onde a função atinge seu valor mínimo.

Definição 1. *Um ponto x^* é um minimizador global de f se $f(x^*) \leq f(x)$, $\forall x \in \mathbb{R}^n$ (ou pelo menos para x no domínio de interesse do modelador).*

Definição 2. *Um ponto x^* é um minimizador local de f se existe uma vizinhança $\mathcal{N}$ de x^* tal que $f(x^*) \leq f(x)$ para $x \in \mathcal{N}$.*

Um ponto que satisfaz esta definição é, às vezes, chamado de *minimizador local fraco*, terminologia dada para distinguir o que chamamos de *minimizador local forte (estrito)*:

Definição 3. *Um ponto x^* é um minimizador local forte de f se existe uma vizinhança $\mathcal{N}$ de x^* tal que $f(x^*) < f(x)$ para $x \in \mathcal{N}$.*

1.1.1 Identificando um mínimo

Pela definição dada acima, devemos examinar todos os pontos na vizinhança de x^* para reconhecer se este é ou não um minimizador local. Se f é regular, existe uma forma mais eficiente de identificar um minimizador local. Em particular, se f é duas vezes continuamente diferenciável, somos capazes de dizer se x^* é um minimizador local, examinando o gradiente $\nabla f(x^*)$ e a Hessiana $\nabla^2 f(x^*)$.

A ferramenta matemática usada para estudar minimizadores de funções regulares é o teorema de Taylor.

Teorema 1. Teorema de Taylor

Suponha que $f : \mathbb{R}^n \longrightarrow \mathbb{R}$ *é continuamente diferenciável e que* $p \in \mathbb{R}^n$. *Então temos que:*

$$f(x+p) = f(x) + \nabla f(x+tp)^T p$$

para algum $t \in (0,1)$. *Além disso, se* f *é duas vezes continuamente diferenciável, temos que:*

$$\nabla f(x+p) = \nabla f(x) + \int_0^1 \nabla^2 f(x+tp)pdt$$

e que

$$f(x+p) = f(x) + \nabla f(x)^T p + \frac{1}{2} p^T \nabla^2 f(x+tp)p$$

para algum $t \in (0,1)$.

Vamos citar as condições necessárias de otimalidade, assumindo que x^* é um minimizador local.

Teorema 2. Condições Necessárias de 1ª Ordem

Se x^* *é um minimizador local e* f *é continuamente diferenciável em uma vizinhança aberta de* x^*, *então* $\nabla f(x^*) = 0$, *ou seja,* x^* *é um ponto estacionário.*

Teorema 3. Condições Necessárias de 2ª Ordem

Se x^* *é um minimizador local de* f *e* $\nabla^2 f$ *é contínua em uma vizinhança aberta de* x^*, *então* $\nabla f(x^*) = 0$ *e* $\nabla^2 f(x^*)$ *é positiva semi-definida.*

Vamos agora descrever condições suficientes, que garantem que x^* é um minimizador local.

Teorema 4. Condições Suficientes de 2ª Ordem

Suponha que $\nabla^2 f$ *é contínua em uma vizinhança aberta de* x^* *e que* $\nabla f(x^*) = 0$ *e* $\nabla^2 f(x^*)$ *é positiva definida. Então* x^* *é um minimizador local forte de* f*.*

1.2 Visão Geral dos Algoritmos

Todos os algoritmos de minimização irrestrita exigem do usuário um ponto inicial, que denotamos por x_0. Iniciando em x_0, o algoritmo gera uma seqüência de iterações $\{x_k\}_{k=0}^{\infty}$ que converge quando um ponto aparentemente se aproxima da solução com uma precisão suficiente. O algoritmo utiliza informações sobre a função f em x_k e possivelmente informações sobre as iterações $x_0, x_1, \ldots, x_{k-1}$, para encontrar uma nova iteração x_{k+1}, onde o valor de f é menor que o valor de f em x_k.

Existem duas formas fundamentais para descrever o movimento de x_k para a nova iteração x_{k+1}, uma delas é o método de busca linear e a outra é o método de região de confiança.

- Método de busca linear:

  Na estratégia de busca linear, a partir de uma direção p_k, busca ao longo dessa direção, começando em x_k, uma nova iteração com um valor da função menor. A distância ao longo de p_k pode ser encontrada resolvendo aproximadamente o seguinte problema de minimização, para encontrar o comprimento de passo α:

$$\underset{\alpha>0}{\text{minimizar}}\, f(x_k + \alpha p_k)$$

- Método de região de confiança:

  Na estratégia de região de confiança, a informação sobre f é usada para construir uma função modelo m_k cujo comportamento próximo ao ponto x_k é semelhante a função objetivo f. Neste método o passo p deverá ser encontrado, resolvendo

aproximadamente o seguinte problema:

$$\underset{p}{\text{minimizar}}\, m_k(x_k + p)$$

onde $x_k + p$ pertence a região de confiança (região em torno de x_k), ou seja, restringimos a busca para um minimizador de m_k em alguma região em torno de x_k.

O modelo m_k é geralmente definido como uma função quadrática da forma:

$$m_k(x_k + p) = f_k + p^T \nabla f_k + \frac{1}{2} p^T B_k p$$

onde f_k, ∇f_k são a função e o gradiente respectivamente calculados em x_k e B_k é, ou a Hessiana $\nabla^2 f_k$, ou alguma aproximação dela.

1.2.1 Método de Newton para Minimização Irrestrita

Vamos discutir o método de Newton para o problema (1).

Se a 1ª e a 2ª derivadas de f estão disponíveis, um modelo quadrático da função objetivo f pode ser obtido a partir da sua expansão em série de Taylor:

$$m_k(x_k + p) = f_k + g_k^T p + \frac{1}{2} p^T H_k p \tag{2}$$

onde f_k, g_k e H_k são, respectivamente, a função, o gradiente e a Hessiana calculados em x_k.

O mínimo do modelo (2) deverá ser alcançado se p_k é um mínimo da função quadrática:

$$\Phi(d) = g_k^T p + \frac{1}{2} p^T H_k p \tag{3}$$

A função Φ tem um ponto estacionário somente se existe um ponto p_k em que o

vetor gradiente se anula, isto é, devemos ter:

$$\nabla\Phi(p_k) = H_k p_k + g_k = 0$$

Conseqüentemente o ponto estacionário p_k de (3) satisfaz o sistema linear:

$$H_k p_k = -g_k \tag{4}$$

Um algoritmo de minimização no qual p_k é definido por (4) é chamado método de Newton e a solução de (4) é chamada direção de Newton.

Algoritmo 1. *Algoritmo de Newton*

Dado x_0,

$k = 0$;

Enquanto $g_k \neq 0$

resolver: $H_k p_k = -g_k$

$x_{k+1} = x_k + p_k$

$k = k + 1$

Fim Enquanto

O algoritmo de Newton não é em geral um algoritmo de descida, podendo divergir. Veremos adiante que se x_0 estiver próximo de um minimizador local x^* com $H(x^*)$ positiva definida, o método converge para x^* rapidamente.

Se $H(x)$ é positiva definida, $\forall x \in \mathbb{R}^n$, então a direção de Newton $p_k = -H_k^{-1} g_k$ é direção de descida:

$$g_k^T p = -g_k^T H_k^{-1} g_k < 0$$

Quando H_k não for positiva definida, a direção de Newton pode não ser definida, pois H_k^{-1} pode não existir e assim uma nova definição de p_k é requerida.

Métodos que utilizam a direção de Newton têm uma taxa de convergência local

rápida, tipicamente quadrática. Quando uma vizinhança da solução é alcançada, a convergência geralmente ocorre em poucas iterações.

1.2.2 Convergência do Método de Newton

Teorema 5. *Suponha que f é duas vezes diferenciável e que a Hessiana $\nabla^2 f(x)$ é Lipschitz contínua em uma vizinhança da solução x^* ($\nabla f(x^*) = 0$ e $\nabla^2 f(x^*)$ é positiva definida). Considere a iteração $x_{k+1} = x_k + p_k$, onde $p_k = -\nabla^2 f_k^{-1} \nabla f_k$. Então:*

1. *se o ponto inicial x_0 é suficientemente próximo de x^*, a seqüência de iterações converge para x^*;*
2. *a razão de convergência de x_k é quadrática; e*
3. *a seqüência das normas do gradiente $\{||\nabla f_k||\}$ converge quadraticamente para zero.*

Prova:

Pela definição do passo de Newton e da condição $\nabla f_* = \nabla f(x^*) = 0$ temos que:

$$\begin{aligned} x_k + p_k - x^* &= x_k - x^* - \nabla^2 f_k^{-1} \nabla f_k \\ &= \nabla^2 f_k^{-1} [\nabla^2 f_k (x_k - x^*) - (\nabla f_k - \nabla f_*)] \end{aligned}$$

Desde que

$$\nabla f_k - \nabla f_* = \int_0^1 \nabla^2 f(x_k + t(x^* - x_k))(x_k - x^*) dt$$

temos

$$\begin{aligned} ||x_k + p_k - x^*|| &\leq \left|\left|\nabla^2 f_k^{-1}\right|\right| \cdot \left|\left|\nabla^2 f_k (x_k - x^*) - (\nabla f_k - \nabla f_*)\right|\right| \\ &\leq \left|\left|\nabla^2 f_k^{-1}\right|\right| \cdot \left|\left|\nabla^2 f_k (x_k - x^*) - \int_0^1 \nabla^2 f(x_k + t(x^* - x_k))(x_k - x^*) dt\right|\right| \\ &= \left|\left|\nabla^2 f_k^{-1}\right|\right| \cdot \left|\left|\int_0^1 \left[\nabla^2 f(x_k) - \nabla^2 f(x_k + t(x^* - x_k))\right](x_k - x^*) dt\right|\right| \\ &\leq \left|\left|\nabla^2 f_k^{-1}\right|\right| \cdot \int_0^1 \left|\left|\nabla^2 f(x_k) - \nabla^2 f(x_k + t(x^* - x_k))\right|\right| \cdot ||x_k - x^*|| dt \end{aligned}$$

$$
\begin{aligned}
&\leq \left|\left|\nabla^2 f_k^{-1}\right|\right| \cdot ||x_k - x^*||^2 \int_0^1 L t dt \\
&\leq 2\left|\left|\nabla^2 f(x^*)^{-1}\right|\right| \cdot ||x_k - x^*||^2 \cdot \frac{1}{2} L \\
&= L\left|\left|\nabla^2 f(x^*)^{-1}\right|\right| \cdot ||x_k - x^*||^2
\end{aligned}
$$

onde L é a constante de Lipschitz de $\nabla^2 f(x)$ para x na vizinhança de x^*. Desde que $\nabla^2 f(x^*)$ é não-singular, e desde que $\nabla^2 f_k \to \nabla^2 f(x^*)$, temos que $\left|\left|\nabla^2 f_k^{-1}\right|\right| \leq 2\,||\nabla^2 f(x^*)^{-1}||$ para todo k suficientemente grande.

Ou ainda:

$$
||x_k + p_k - x^*|| \leq L\left|\left|\nabla^2 f(x^*)^{-1}\right|\right| \cdot ||x_k - x^*||^2 = \widetilde{L}||x_k - x^*||^2
$$

onde $\widetilde{L} = L\,||\nabla^2 f(x^*)^{-1}||$. Usando esta desigualdade indutivamente deduzimos que se o ponto inicial está suficientemente próximo de x^*, então a seqüência converge para x^*, e a taxa de convergência é quadrática.

Utilizando a relação $x_{k+1} - x_k = p_k$ e $\nabla f_k + \nabla^2 f_k p_k = 0$, obtemos que:

$$
\begin{aligned}
||\nabla f(x_{k+1})|| &= \left|\left|\nabla f(x_{k+1}) - \nabla f_k - \nabla^2 f(x_k) p_k\right|\right| \\
&= \left|\left|\int_0^1 \nabla^2 f(x_k + t p_k)(x_{k+1} - x_k) dt - \nabla^2 f(x_k) p_k\right|\right| \\
&\leq \int_0^1 \left|\left|\nabla^2 f(x_k + t p_k) - \nabla^2 f(x_k)\right|\right| \cdot ||p_k|| dt \\
&\leq \frac{1}{2} L ||p_k||^2 \\
&\leq \frac{1}{2} L \left|\left|\nabla^2 f(x_k)^{-1}\right|\right|^2 \cdot ||\nabla f_k||^2 \\
&\leq 2L \left|\left|\nabla^2 f(x^*)^{-1}\right|\right|^2 \cdot ||\nabla f_k||^2
\end{aligned}
$$

provando que a norma do gradiente converge para 0 quadraticamente. ■

1.2.3 Outros Métodos

No método de Newton é necessário o cálculo da Hessiana H_k. O cálculo explícito dessa matriz deixa, às vezes, o processo caro e até mesmo propício ao erro. Uma das alternativas, que não requer o cálculo da Hessiana, é utilizar a direção de busca Quase-Newton, calculando a direção através de uma aproximação B_k da Hessiana. A aproximação B_k é atualizada em cada iteração e satisfaz a seguinte condição:

$$B_{k+1}s_k = y_k$$

onde

$$s_k = x_{k+1} - x_k \text{ e } y_k = \nabla f_{k+1} - \nabla f_k.$$

As fórmulas mais populares para atualizar B_k, são a fórmula *simétrica de posto um*, definida por:

$$B_{k+1} = B_k + \frac{(y_k - B_k s_k)(y_k - B_k s_k)^T}{(y_k - B_k s_k)^T s_k}$$

e a fórmula *BFGS*, (Broyden, Fletcher, Goldfarb, e Shanno) que é definida por:

$$B_{k+1} = B_k - \frac{B_k s_k s_k^T B_k}{s_k^T B_k s_k} + \frac{y_k y_k^T}{y_k^T s_k}$$

A direção de busca Quase-Newton é então definida como:

$$p_k = -B_k^{-1}\nabla f_k$$

Teorema 6. *Suponha que f é duas vezes continuamente diferenciável, que as iterações geradas pelo algoritmo BFGS convergem para um minimizador x^* e que a Hessiana $\nabla^2 f(x)$ é Lipschitz contínua em uma vizinhança da solução x^*. Suponha também que $\sum_{k=1}^{\infty} ||x_k - x^*|| < \infty$. Então x_k converge para x^* em uma razão superlinear.*

Prova: Ver [3].

Consideremos agora as direções de busca geradas pelos métodos dos gradientes conjugados não-lineares. Eles têm a forma:

$$p_k = -\nabla f(x_k) + \beta_k p_{k-1}$$

onde β_k é um escalar que garante que p_k e p_{k-1} são conjugados.

Estes métodos não alcançam a rápida taxa de convergência dos métodos de Newton e Quase-Newton, mas têm a vantagem de não exigir o armazenamento de matrizes.

2 Minimização Restrita

Minimização restrita consiste em minimizar uma função objetivo, sujeita a um conjunto de restrições impostas nas variáveis. Em termos matemáticos:

$$\begin{aligned} &\underset{x\in\mathbb{R}^n}{\text{minimizar}}\, f(x) \\ &\text{sujeito a} \quad c_i(x) = 0,\ i = 1, 2, \ldots, m' \\ &\qquad\qquad\ \ c_i(x) \geq 0,\ i = m'+1, \ldots, m \end{aligned} \tag{5}$$

onde a função objetivo f e as funções restrições c_i são funções escalares de valor real.

Um ponto $\widehat{x}$ é dito viável se satisfaz todas as restrições de (5). O conjunto de todos os pontos viáveis é chamado de região viável.

2.1 Restrições Lineares de Igualdade

Consideremos o seguinte problema:

$$\begin{aligned} &\underset{x\in\mathbb{R}^n}{\text{minimizar}}\, f(x) \\ &\text{sujeito a } Ax = b \end{aligned} \tag{6}$$

onde f é duas vezes continuamente diferenciável, a matriz A é $t \times n$ e a linha i de A contém os coeficientes correspondentes a restrição i.

Um ponto viável x^* é um mínimo local de (6) se $f(x^*) \leq f(x)$, para todo ponto viável x em alguma vizinhança de x^*. Vamos caracterizar o conjunto de todos os pontos viáveis em uma vizinhança de um ponto viável, para determinar as condições de otimalidade para (6).

Devemos assumir que b pertença ao espaço coluna de A, pois caso contrário as restrições seriam inconsistentes e não haveria ponto viável. Como as restrições formam um sistema linear, as propriedades de subespaço linear tornam possível especificar uma caracterização de todas as mudanças viáveis de um ponto viável. Seja p o passo entre

2 pontos viáveis $\overline{x}$ e $\widehat{x}$, então pela linearidade de A temos:

$$A(\overline{x} - \widehat{x}) = A\overline{x} - A\widehat{x} = b - b = 0$$

Portanto $Ap = 0$, ou seja, o passo p de um ponto viável para qualquer outro ponto viável deve ser ortogonal às linhas de A. Qualquer passo de um ponto viável ao longo de p não viola as restrições, desde que $A(\widehat{x} + \alpha p) = A\widehat{x} + \alpha Ap = A\widehat{x} = b$.

A direção p é chamada direção viável com respeito as restrições de (6).

Seja Z a matriz cujas colunas formam uma base para o subespaço de vetores que satisfazem $Ap = 0$. Segue que se p satisfaz $Ap = 0$, então p pode ser escrito como Zp_z para algum vetor p_z.

2.1.1 Condições de Otimalidade

Expandindo f em série de Taylor sobre x^* ao longo de uma direção viável p, ($p = Zp_z$) podemos determinar a otimalidade do ponto viável x^* dado:

$$f(x^* + \epsilon p) = f(x^*) + \epsilon p^T \nabla f(x^*) + \frac{1}{2}\epsilon^2 p^T \nabla^2 f(x^* + \epsilon\theta p)p$$

onde θ satisfaz $0 \leq \theta \leq 1$ e ϵ é um escalar positivo, ou ainda, como $p = Zp_z$, tomando $g(x) = \nabla f(x)$ e $G(x) = \nabla^2 f(x)$, temos:

$$f(x^* + \epsilon Zp_z) = f(x^*) + \epsilon p_z^T Z^T g(x^*) + \frac{1}{2}\epsilon^2 p_z^T Z^T G(x^* + \epsilon\theta p)Zp_z \tag{7}$$

Suponha que x^* é um mínimo local. Se $g(x^*) \neq 0$, então existe um vetor p para qual

$$p_z^T Z^T g(x^*) < 0 \tag{8}$$

(p pode ser pego como $-g(x^*)$)

Qualquer vetor p que satisfaz (8) é chamado direção de descida em x^*. Dada qualquer direção de descida p, existe um escalar positivo $\overline{\epsilon}$ tal que para todo ϵ positivo

satisfazendo $\epsilon \leq \overline{\epsilon}$, temos que $\epsilon p_z^T Z^T g(x^*) + \frac{1}{2}\epsilon^2 p_z^T Z^T G(x^* + \epsilon\theta p) Z p_z < 0$. Por (7), segue que $f(x^* + \epsilon p) < f(x^*)$ para tal ϵ, contradizendo o fato de x^* ser o mínimo local.

Portanto, uma condição necessária para x^* ser um mínimo local de (6) é que $p_z^T Z^T g(x^*) = 0$ para todo p_z, o que implica que $Z^T g(x^*) = 0$. O vetor $Z^T g(x^*)$ é chamado o gradiente projetado de f calculado em x^*.

Assumindo x^* um mínimo local de (6), desde que $Z^T g(x^*) = 0$, a expansão em série de Taylor (7) torna-se:

$$f(x^* + \epsilon Z p_z) = f(x^*) + \frac{1}{2}\epsilon^2 p_z^T Z^T G(x^* + \epsilon\theta p) Z p_z \tag{9}$$

para algum θ, $0 \leq \theta \leq 1$ e ϵ escalar positivo.

Se $G(x^*)$ é indefinida, pela continuidade, G deverá ser indefinida para todos os pontos em alguma vizinhança de x^*, e podemos escolher ϵ pequeno o suficiente para que $x^* + \epsilon p$ esteja nessa vizinhança. Escolhendo p tal que $p^T G(x^* + \epsilon\theta p) p < 0$ temos:

$$f(x^* + \epsilon Z p_z) - f(x^*) < 0 \Longrightarrow f(x^* + \epsilon Z p_z) < f(x^*)$$

o que contradiz a otimalidade de x^*.

A matriz $Z^T G(x^*) Z$, que é chamada matriz Hessiana projetada, deverá ser positiva semi-definida. Portanto as condições para x^* ser um mínimo local de (6), são:
Condições necessárias para um mínimo de (6):

A1. $Ax^* = b$;

A2. $Z^T g(x^*) = 0$;

A3. $Z^T G(x^*) Z$ é positiva semi-definida.

Condições suficientes para um mínimo de (6):

B1. $Ax^* = b$;

B2. $Z^T g(x^*) = 0$;

B3. $Z^T G(x^*) Z$ é positiva definida.

Para verificar que essas condições são suficientes, observe que a condição B2. é necessária e a expansão de f sobre x^* é portanto dada por (9). Se $G(x^*)$ é positiva definida, pela continuidade, G é positiva definida para todos os pontos em alguma vizinhança de x^*. Para ϵ suficientemente pequeno (tal que $x+\epsilon p$ esteja na vizinhança), e para qualquer direção p, é válido que $p^T G(x^* + \epsilon p)p > 0$. Por (9), isto implica que:

$$f(x^* + \epsilon Z p_z) - f(x^*) > 0 \Longrightarrow f(x^* + \epsilon Z p_z) > f(x^*)$$

2.2 Desenvolvimento dos Algoritmos

Seja Y a matriz cujas colunas formam uma base para o espaço coluna de A^T. Desde que Y e Z definem espaços complementares, todo vetor $x \in \mathbb{R}^n$ tem uma única expansão como uma combinação linear das colunas de Y e Z:

$$x = Y x_y + Z x_z$$

para algum vetor $x_y \in \mathbb{R}^t$ (a porção espaço coluna de x) e um vetor $x_z \in \mathbb{R}^{n-t}$ (a porção espaço nulo de x).

Seja x^* solução de (6), dada por $x^* = Y x_y^* + Z x_z^*$, então:

$$b = Ax^* = A(Y x_y^* + Z x_z^*) = AY x_y^* + AZ x_z^* = AY x_y^*$$

Agora, $x_y \in \mathbb{R}^t$ é unicamente determinado, pois por definição de Y, a matriz AY é não-singular. Portanto as restrições determinam a porção espaço coluna da solução de (6). De fato, veremos que calcular a solução do problema restrito (6), pode ser visto como um problema irrestrito nas $n - t$ variáveis x_z.

Exemplo 1. Considere a seguinte restrição:

$$x_1 + x_2 + x_3 = 3$$

para qual $A = \begin{pmatrix} 1 & 1 & 1 \end{pmatrix}$. A matriz Y pode ser tomada como A^T e uma possível matriz Z é:

$$Z = \begin{pmatrix} -\frac{\sqrt{3}}{3} & -\frac{\sqrt{3}}{3} \\ \frac{3+\sqrt{3}}{6} & -\frac{3-\sqrt{3}}{6} \\ -\frac{3-\sqrt{3}}{6} & \frac{3+\sqrt{3}}{6} \end{pmatrix}$$

Substituindo em $AYx_y^* = b$, nós obtemos $x_y^* = 1$. Portanto, independente da função f, a solução deverá ser da forma:

$$x^* = \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix} + \begin{pmatrix} -\frac{\sqrt{3}}{3} & -\frac{\sqrt{3}}{3} \\ \frac{3+\sqrt{3}}{6} & -\frac{3-\sqrt{3}}{6} \\ -\frac{3-\sqrt{3}}{6} & \frac{3+\sqrt{3}}{6} \end{pmatrix} x_z^*$$

2.2.1 Algoritmo Modelo

Como observado anteriormente na seção (2.1), o passo p de qualquer ponto viável para qualquer outro ponto viável, deve ser ortogonal as linhas de A. Portanto, se x_k é viável, p_k (a busca direcional na direção k) deverá satisfazer:

$$Ap_k = 0 \tag{10}$$

Agora, a seguinte equivalência é válida:

$$Ap_k = 0 \Longleftrightarrow p_k = Zp_z \tag{11}$$

para algum vetor $p_z \in \mathbb{R}^{n-t}$, pois qualquer vetor p_z que satisfaz (10) deverá ser uma combinação das colunas de Z.

Podemos então desenvolver um algoritmo modelo para resolver (6), generalizando uma seqüência de iterações viáveis:

Dado um ponto inicial viável x_0, seja $k = 0$, repita os seguintes passos:

Algoritmo *Algoritmo modelo*

(i) (Teste da Convergência) x_k é solução se as condições para convergência são satisfeitas.

(ii) (Direção de Busca) Calcular um vetor $p_z \in \mathbb{R}^{n-t}$, $p_z \neq 0$. A direção de busca é:

$$p_k = Zp_z$$

(iii) Atualizar: $x_{k+1} = x_k + p_k$, $k = k + 1$ e ir para o passo (i).

2.2.2 Método de Newton para Minização Restrita

O método de Newton descrito na seção (1.2.1) pode ser adaptado para minimização restrita. A direção de Newton para (6) é a direção que resolve o problema restrito:

$$\begin{aligned} &\underset{p\in\mathbb{R}^n}{\text{minimizar}}\ f_k + g_k^T p + \frac{1}{2}p^T G_k p \\ &\text{sujeito a} \quad Ap = 0 \end{aligned} \tag{12}$$

Se G_z denota a matriz Hessiana projetada $Z^T G_k Z$, e g_z denota o gradiente projetado $Z^T g_k$, a solução de (12) é dada por Zp_z, onde p_z é solução de

$$G_z p_z = -g_z \tag{13}$$

O algoritmo de Newton para minimização restrita fica então:

Algoritmo 1. *Algoritmo de Newton para minimização restrita*

Dado x_0,

$k = 0$;

Calcular Z (base para o espaço nulo de A)

Enquanto $||g_z|| > \epsilon$

$g_z = Z^T g_k$

$G_z = Z^T G_k Z$

resolver: $G_z p_z = -g_z$

$x_{k+1} = x_k + Z p_z$

$k = k + 1$

Fim Enquanto

Exemplo 2. Considere o problema:

$$\begin{aligned} &\underset{x\in\mathbb{R}^3}{\text{minimizar}} \quad x_1^2x_2^3 + 4x_1^2x_3^2 + x_2^4x_3^2 + 3x_1x_2 + 4x_2x_3 + 5x_1x_3 + x_1x_2x_3 \\ &\text{sujeito a} \quad x_1 + x_2 + x_3 = 3 \end{aligned}$$

Para este problema $A = \begin{pmatrix} 1 & 1 & 1 \end{pmatrix}$ e a matriz Z pode ser tomada como:

$$Z = \begin{pmatrix} -\frac{\sqrt{3}}{3} & -\frac{\sqrt{3}}{3} \\ \frac{3+\sqrt{3}}{6} & -\frac{3-\sqrt{3}}{6} \\ -\frac{3-\sqrt{3}}{6} & \frac{3+\sqrt{3}}{6} \end{pmatrix}$$

No ponto viável $x_0 = (-1, 5, -1)^T$, a solução de (13) é

$$p_z = \begin{pmatrix} 0.72450 \\ -0.32483 \end{pmatrix}$$

Se G_z é positiva definida e p_z é a solução de (13), a direção de Newton Zp_z é uma direção de descida, desde que:

$$g_k^T Z p_z = ((Z^T)^{-1} g_z)^T Z p_z = g_z^T Z^{-1} Z p_z = g_z^T p_z$$

Agora, $G_z p_z = -g_z \Longrightarrow p_z = -G_z^{-1} g_z$, então:

$$g_k^T Z p_z = g_z^T p_z = -g_z G_z^{-1} g_z < 0$$

Se G_z não é positiva definida, a aproximação quadrática local não é limitada inferiormente, ou não tem um único mínimo. Neste caso, é necessário modificar a definição de p_z.

2.2.3 Representação do espaço nulo das restrições

Para garantir que as restrições são satisfeitas em cada iteração, precisamos encontrar a matriz Z. A matriz Z que procuramos é a matriz cujas colunas formam uma base para o espaço nulo de A. Como veremos na seção (3.2), basta tomar Z como sendo as $n - t$ colunas da matriz Q' (matriz da fatoração $Q'R'$ de A^T).

Uma vantagem fundamental para usarmos a fatoração QR é que a escolha de Z não causa uma deterioração no condicionamento quando resolvemos o problema (6). Em um método tipo Newton, podemos ver que $cond(G_z)$ (número condição de G_z) depende dos números condição de Z e de G_k:

Como $G_z = Z^T G_k Z$ então:

$$cond(G_z) = cond(Z^T G_k Z) \leq cond(G_k)(cond(Z))^2$$

A matriz Z determinada pela fatoração QR tem um número condição igual a 1, e portanto:

$$cond(G_z) \leq cond(G_k)$$

o que implica que o condicionamento do problema original não é piorado pelo método numérico.

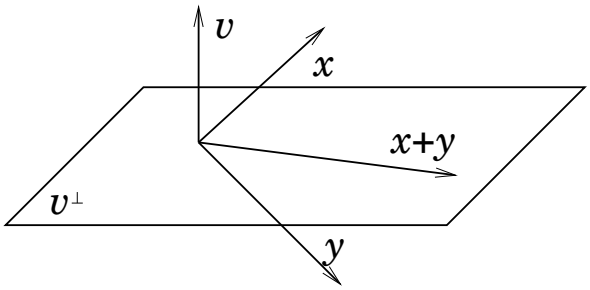
3 Fatoração QR

3.1 Fatoração QR

A partir de uma seqüência de matrizes elementares ortogonais, podemos reduzir uma matriz $A \in \mathbb{R}^{m \times n}$ de posto n, para a forma triangular superior.

Podemos introduzir elementos nulos na matriz A, da seguinte maneira: dado um vetor arbitrário $x \in \mathbb{R}^n$ e um vetor fixo $v \in \mathbb{R}^n$, obtemos o vetor y, reflexão de x em $v^{\perp}$.

Geometricamente:



Como v está na direção de $y - x$, temos $y - x = \alpha v$, $\alpha \in \mathbb{R}$ e também $x + y \in v^{\perp}$, então:

$$v^T(x + y) = 0$$

$$v^T(x + x + \alpha v) = 0$$

$$v^T(2x + \alpha v) = 0$$

$$2v^T x + \alpha v^T v = 0$$

$$\alpha = -\frac{2v^T x}{v^T v}$$

Portanto:

$$y = x - \frac{2v^T x}{v^T v} \cdot v$$

Obtemos a reflexão em forma de transformação linear (matriz):

$$y = x - \frac{2v^T x}{v^T v} \cdot v = x - \frac{2vv^T}{v^T v} \cdot x = \left(I - \frac{2vv^T}{v^T v} \right) \cdot x = Px$$

em que P é chamada matriz de Householder.

Vamos utilizar a transformação de Householder para introduzir elementos nulos na matriz A. O sucesso depende do seguinte resultado:

Proposição 1. *Suponha que y é o vetor coluna $(1, 0, 0, 0)$, $\gamma = ||x||$, e $v = x + \gamma y$. Então $Px = -\gamma y = (-\gamma, 0, 0, 0)$.*

Prova:

Suponha $x = (x_1, x_2, \ldots, x_n)$ e $Px = x - \dfrac{2vv^T}{v^T v} \cdot x$

Agora $v^T x = (x + \gamma y)^T x = x^T x + \gamma y^T x = x^T x + \gamma x_1$ e

$$\begin{aligned} v^T v &= (x + \gamma y)^T (x + \gamma y) \\ &= x^T x + 2\gamma x^T y + \gamma^2 y^T y \\ &= x^T x + 2\gamma x_1 + \gamma^2 \\ &= x^T x + 2\gamma x_1 + x^T x \\ &= 2(x^T x + \gamma x_1) \end{aligned}$$

então

$$Px = x - \frac{2vv^T}{v^T v} x = x - 2\frac{(x + \gamma y)(x^T x + \gamma x_1)}{2(x^T x + \gamma x_1)} = x - x - \gamma y = -\gamma y = (-\gamma, 0, 0, 0). \blacksquare$$

A propriedade de reflexão das matrizes de Householder, implica que uma seqüência de n matrizes $\{P_i\}$, $i = 1, \ldots, n$, podem ser aplicadas do lado esquerdo de uma matriz $A \in \mathbb{R}^{m \times n}$ de posto n, para reduzi-la à forma triangular superior.

Em particular, P_1 é construída para transformar a primeira coluna a_1 de A em um

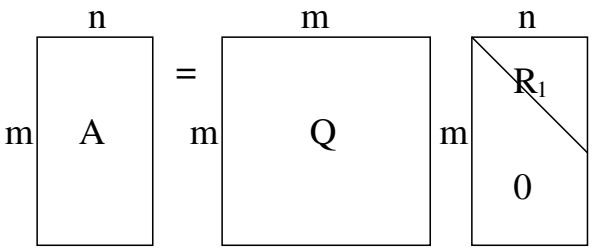
múltiplo de e_1. Após n transformações, nós obtemos:

$$P_n P_{n-1} \dots P_2 P_1 A = Q^T A = \begin{pmatrix} R_1 \\ 0 \end{pmatrix} = R \tag{14}$$

onde $R_1 \in \mathbb{R}^{n \times n}$ é uma matriz triangular superior, não-singular e Q^T é o produto $P_n P_{n-1} \dots P_2 P_1$. A forma (14) é chamada fatoração QR de A.

3.2 Relação da fatoração QR com os Espaços Fundamentais

Seja $A \in \mathbb{R}^{m \times n}$, $m \geq n$, com posto n (A tem posto completo). Existe uma matriz ortogonal $Q \in \mathbb{R}^{m \times m}$ e $R \in \mathbb{R}^{m \times n}$ tal que $A = QR$.



$R = \begin{pmatrix} R_1 \\ 0 \end{pmatrix}$, em que $R_1 \in \mathbb{R}^{n \times n}$ é triangular superior e não-singular.

Sejam $\{a_j\}$, $j = 1, \dots, n$ as colunas da matriz A, $\{q_i\}$, $i = 1, \dots, m$ as colunas da matriz Q e $\{r_{ij}\}$, $i = 1, \dots, m$ e $j = 1, \dots, n$ os elementos da matriz R. Pela fatoração QR temos que:

$$\begin{aligned} a_1 &= r_{11} q_1 \\ a_2 &= r_{12} q_1 + r_{22} q_2 \\ &\vdots \quad\quad \vdots \\ a_n &= r_{1n} q_1 + r_{2n} q_2 + \dots + r_{nn} q_n \end{aligned}$$

Observe que as colunas de A são combinações lineares das n primeiras colunas de Q, e portanto $\mathcal{R}(A) \subset \mathcal{R}(Q)$ ($\mathcal{R}(A)$ é o espaço coluna de A).

Seja $Q = \begin{pmatrix} Q_1 & Q_2 \end{pmatrix}$, onde $Q_1 \in \mathbb{R}^{m\times n}$ e $Q_2 \in \mathbb{R}^{m\times(m-n)}$, temos que $\mathcal{R}(A) = \mathcal{R}(Q_1)$, ou seja, Q_1 é base ortogonal para $\mathcal{R}(A)$ e Q_2 é base ortogonal para $\mathcal{N}(A^T)$ ($\mathcal{N}(A^T)$ é o espaço nulo de A^T).

Se encontrarmos a fatoração $Q'R'$ da matriz $A^T \in \mathbb{R}^{n\times m}$ teremos: $A^T = Q'R'$, onde $Q' \in \mathbb{R}^{n\times n}$ e $R' \in \mathbb{R}^{n\times m}$ e da mesma maneira como visto acima, sendo $Q' = \begin{pmatrix} Q'_1 & Q'_2 \end{pmatrix}$ onde $Q'_1 \in \mathbb{R}^{n\times m}$ e $Q'_2 \in \mathbb{R}^{n\times(n-m)}$. Temos que Q'_1 é base ortogonal para $\mathcal{R}(A^T)$ (o espaço coluna de A^T) e Q'_2 é base ortogonal para $\mathcal{N}(A)$ (o espaço nulo de A).

4 Implementação no ambiente CUTEr

Para implementação computacional da metodologia descrita nos capítulo anteriores, utilizamos o ambiente CUTE (Constrained and Unconstrained Testing Environment) desenvolvido por N.I.M. Gould, D. Orban, e Ph. L. Toint.

4.1 O Ambiente CUTEr

O CUTEr é uma revisão do CUTE, é um ambiente utilizado para testar problemas de otimização e de álgebra linear. O ambiente contém uma coleção de problemas testes, que juntamente com o Fortran 77, Fortran 90 e ferramentas do Matlab, auxiliam os autores de softwares, na verificação, comparação e melhoramento de novos e já existentes pacotes de otimização.

Para selecionar os problemas adequados ao nosso estudo, basta olharmos para a sua classificação. Cada problema é classificado da seguinte maneira:

$$X_1X_2X_3r - X_4X_5 - n - m$$

onde:

- X_1 indica o tipo da função objetivo, que pode receber o valor **N** se nenhuma função objetiva for definida, **C** se a função é constante, **L** se é linear, **Q** se é quadrática, **S** se é uma soma de quadrados e **O** se a função objetivo é nenhuma das funções anteriores;

- X_2 indica o tipo de restrição, recebe o valor **U** se o problema é irrestrito, **X** se as restrições são variáveis fixas, **B** se as restrições são limitadas nas variáveis fixas, **N** se as restrições representam a matriz de um grafo, **L** se são restrições lineares, **Q** se são restrições quadráticas e **O** se as restrições são mais gerais que qualquer uma das anteriores;

- X_3 indica a suavidade, recebe o valor **R** se o problema é regular e **I** se o problema é irregular;

- $r = 0$, 1 ou 2 indica o grau de maior derivada fornecida;
- X_4 indica a origem e/ou a utilidade do problema, recebendo o valor **A** se o problema for acadêmico, ou seja, foi construído por pesquisadores para testar algoritmos, **M** se o problema é parte de um exercício de modelagem, onde os valores reais da solução não são usados em uma aplicação prática e **R** se a solução do problema é usada em aplicações reais para outros propósitos que não sejam testar algoritmos;
- X_5 indica se o problema contém ou não variáveis internas explícitas, recebendo o valor **Y** se o problema contém variáveis internas explícitas e **N** no caso contrário;
- n indica o número de variáveis, podendo ser igual a V, indicando que o número de variáveis pode ser escolhido pelo usuário; e
- m indica o número de restrições, podendo também ser escolhido pelo usuário se $m = V$.

Toda a estrutura, instalação, organização das ferramentas, interfaces do ambiente poderá ser encontrada em [8].

4.2 Testes Numéricos

A seguir estão o programa principal e a sub-rotina utilizada para fazer os testes numéricos, ambos implementados no ambiente CUTEr com a ajuda do Fortran 77:

```
1
2         PROGRAM GRASIMA
3
4   C     DECLARAÇÃO DAS VARIÁVEIS
5
6         PARAMETER(NMAX=200,MMAX=200)
7         PARAMETER (INPUT=47,IOUT=6)
8         INTEGER NU,M,INPUT,IOUT
9         DOUBLE PRECISION X(NMAX),F,H(NMAX,NMAX),A(MMAX,NMAX)
10        DOUBLE PRECISION BU(NMAX),BL(NMAX),CU(MMAX),CL(MMAX)
11        DOUBLE PRECISION G(NMAX),V(MMAX),AT(NMAX,MMAX)
12        CHARACTER*10 PNAME,XNAMES(NMAX),GNAMES(MMAX)
13
14
15        OPEN (INPUT, FILE ='OUTSDIF.d',FORM = 'FORMATTED',
16       .       STATUS = 'OLD')
17        REWIND INPUT
18
19  C     Inicialização dos Dados
20        CALL CSETUP(INPUT,IOUT,NU,M,X,BL,BU,NMAX,EQUATN,LINEAR,
21       *            V,CL,CU,MMAX,.FALSE.,.FALSE.,.FALSE.)
22
23        CALL CNAMES(NU,M,PNAME,XNAMES,GNAMES)
24
25  C     Cálculo da Matriz de Restrições
26        CALL CGR(NU,M,X,.FALSE.,M,V,G,.FALSE.,MMAX,NMAX,A)
27
28        CALL NEWTON(M,NU,A,NMAX,MMAX,X,F,G)
29
30        CLOSE(INPUT)
31
32        WRITE(*,*) 'VETOR X SOLUÇÃO = ',(X(I),I=1,NU)
33
34        STOP
35        END
```

Nas linhas 24, 27 e 30 utilizamos as sub-rotinas do CUTEr, CSETUP, CNAMES e CGR, para obter, respectivamente, os dados do problema; o nome do problema, das variáveis e das restrições; e o gradiente das restrições. Como nossas restrições são restrições lineares, o gradiente das restrições é utilizado para formar a matriz das restrições.

```
1
2         SUBROUTINE NEWTON(M,NU,A,NMAX,MMAX,X,F,G)
3
4         DOUBLE PRECISION ONE,ZERO
5         PARAMETER(ONE=1.0D+0,ZERO=0.0D+0,EPS=0.00001)
6         INTEGER NU,NZ,M,K,NMAX,MMAX,IPIV(NMAX),INFO
7         DOUBLE PRECISION X(NMAX),F,H(NMAX,NMAX)
8         DOUBLE PRECISION DNRM2,NPROJE,P(NMAX),PN(NMAX)
9         DOUBLE PRECISION TAU(MMAX),WORK(NMAX),G(NMAX),V(MMAX),Y(NMAX)
10        DOUBLE PRECISION A(MMAX,NMAX),AT(NMAX,MMAX),Z(NMAX,MMAX)
11        DOUBLE PRECISION C(NMAX,NMAX),D(NMAX,NMAX)
12        REAL EPS
13        CHARACTER*1 T,n
14
15  C     Transposição da matriz A
16        DO I=1,NU
17           DO J=1,M
18           AT(I,J)=A(J,I)
19           END DO
20        END DO
21
22  C     Cálculo da fatoração QR de AT
23        CALL DGEQRF(NU,M,AT,NMAX,TAU,WORK,NU,INFO)
24
25  C     Gera a matriz Q de AT
26        CALL DORGQR(NU,NU,M,AT,NMAX,TAU,WORK,NU,INFO)
27
28  C     Extrai as últimas N-M colunas da Q
29        DO I=1,NU
30           DO J=1,NU
31              Z(I,J)=AT(I,J+M)
32           END DO
33        END DO
34
35        K=0
36
37  C     Cálculo do gradiente(G) da função objetivo
38        CALL COFG(NU,X,F,G,.TRUE.)
39
40  C     Cálculo do produto Y=Z^T*G (gradiente projetado)
41        NZ=NU-M
42        CALL DGEMV('T',NU,NZ,ONE,Z,NMAX,G,1,ZERO,Y,1)
43
44  C     Cálculo da norma do gradiente projetado
45        NPROJE=DNRM2(NZ,Y,1)
```

```
46
47  10    IF (NPROJE.GT.EPS) THEN
48
49  C     Cálculo da hessiana(H) da função objetivo
50        CALL CDH(NU,M,X,MMAX,V,NMAX,H)
51
52  C     Cálculo do produto (matriz*matriz) C=H*Z
53        CALL DGEMM('n','n',NU,NZ,NU,ONE,H,NMAX,Z,NMAX,ZERO,C,NMAX)
54
55  C     Cálculo do produto (matriz*matriz) D=Z^T*C
56        CALL DGEMM('T','n',NZ,NZ,NU,ONE,Z,NMAX,C,NMAX,ZERO,D,NMAX)
57
58        DO I=1,NZ
59          P(I)=(-1)*Y(I)
60        END DO
61
62  C     Resolve  D*P=-Y usando a LU de D
63        CALL DGESV(NZ,1,D,NMAX,IPIV,P,NMAX,INFO)
64
65  C     Cálculo do produto PN=Z*P (Z * direçao)
66        CALL DGEMV('n',NU,NZ,ONE,Z,NMAX,P,1,ZERO,PN,1)
67
68  C     Atualiza o valor de X
69        CALL DAXPY(NU,ONE,PN,1,X,1)
70
71  C     Cálculo do gradiente(G) da função objetivo
72        CALL COFG(NU,X,F,G,.TRUE.)
73
74  C     Cálculo do produto Y=Z^T*G (gradiente projetado)
75        CALL DGEMV('T',NU,NZ,ONE,Z,NMAX,G,1,ZERO,Y,1)
76
77  C     Cálculo da norma do gradiente projetado
78        NPROJE=DNRM2(NZ,Y,1)
79
80        K=K+1
81        GO TO 10
82        END IF
83
84        RETURN
85        END
```

Os passos 15 até 33 foram feitos para se calcular a matriz Z, matriz cujas colunas são base para o espaço nulo de A^T. Nas linhas 38, 42 e 45, calculamos, respectivamente, o gradiente, o gradiente projetado e a norma do gradiente projetado. Se a norma

do gradiente projetado é maior que $EPS = 10^{-5}$, então calculamos a Hessiana da função objetivo (linha 49), a Hessiana projetada (linhas 52 à 56) e resolvemos o sistema $Z^T H_k Z p_k = -Z^T g_k$ (linha 62). Atualizamos o valor de x na linha 60 ($x_{k+1} = x_k + Z p_k$). Calculamos o valor do gradiente em x_{k+1} (linha 72), o valor do gradiente projetado (linha 75), a norma do gradiente projetado (linha 78), e retornamos à linha 47. Quando a condição da linha 47 não é satisfeita, o algoritmo retorna a solução do problema.

Os problemas que vamos testar devem ter $X_2 = L$ (restrições lineares), $X_3 = R$ (devem ser problemas regulares) e $r = 2$ (precisamos da Hessiana da função objetivo). Escolhemos alguns problemas e listamos abaixo os testes feito com o Método de Newton. Observe que para as funções quadráticas (que é o caso dos problemas BT3, GENHS28, HS28, HS48, HS51, HS52), o Método de Newton converge em apenas 1 iteração. Na tabela, mostramos a norma do gradiente projetado ($Z^T g_k$) em cada iteração:

Problema	Classificação	Número de variáveis	Número de restrições	Norma do gradiente projetado
BT3	SLR2-AY-5-3	5	3	68.7078542 7.67864796^{-15}
GENHS28	QLR2-AY-10-8	10	8	7.74301427 1.93928011^{-15}
HS28	SLR2-AY-3-1	3	1	7.46420027 2.05144633^{-15}
HS48	SLR2-AY-5-2	5	2	25.0421866 9.9213059^{-15}
HS49	OLR2-AY-5-2	5	2	102.563376 29.9777823 8.8820725 2.63172519 0.779770425 0.231043089 0.0684572116 0.0202836182 0.00600996096 0.00178072917 0.000527623459 0.000156332877 4.63208524^{-05} 1.3724697^{-05} 4.06657689^{-06}
HS50	OLR2-AY-5-3	5	3	803.346365 200.014589 60.0770101 18.3142867 5.71827265 1.79062217 0.428805518 0.018361132 1.65352699^{-06}
HS51	QLR2-AY-5-3	5	3	6.34580419 2.54216271^{-15}
HS52	QLR2-AY-5-3	5	3	39.9634448 1.01194492^{-15}

Implementamos também o método BFGS visto na seção (1.2.3) utilizando a seguinte fórmula:

$$B_{k+1} = B_k - \frac{B_k s_k s_k^T B_k}{s_k^T B_k s_k} + \frac{y_k y_k^T}{y_k^T s_k}$$

A seguir estão o programa principal e a sub-rotina utilizada nos testes numéricos.

```
1
2         PROGRAM BFGSMA
3
4   C     DECLARAÇÃO DAS VARIÁVEIS
5
6         PARAMETER(NMAX=200,MMAX=200)
7         PARAMETER (INPUT=47,IOUT=6)
8         INTEGER NU,M,INPUT,IOUT
9         DOUBLE PRECISION X(NMAX),F,H(NMAX,NMAX),A(MMAX,NMAX)
10        DOUBLE PRECISION BU(NMAX),BL(NMAX),CU(MMAX),CL(MMAX)
11        DOUBLE PRECISION G(NMAX),V(MMAX),AT(NMAX,MMAX)
12        CHARACTER*10 PNAME,XNAMES(NMAX),GNAMES(MMAX)
13
14
15        OPEN (INPUT, FILE ='OUTSDIF.d',FORM = 'FORMATTED',
16       .       STATUS = 'OLD')
17        REWIND INPUT
18
19  C     Inicialização dos Dados
20        CALL CSETUP(INPUT,IOUT,NU,M,X,BL,BU,NMAX,EQUATN,LINEAR,
21       *            V,CL,CU,MMAX,.FALSE.,.FALSE.,.FALSE.)
22
23        CALL CNAMES(NU,M,PNAME,XNAMES,GNAMES)
24
25  C     Cálculo da Matriz de Restrições
26        CALL CGR(NU,M,X,.FALSE.,M,V,G,.FALSE.,MMAX,NMAX,A)
27
28        CALL BFGS(M,NU,A,NMAX,MMAX,X,F,G)
29
30        CLOSE(INPUT)
31
32        WRITE(*,*) 'VETOR X SOLUÇÃO = ',(X(I),I=1,NU)
33
34        STOP
35        END
```

```
      SUBROUTINE BFGS(M,NU,A,NMAX,MMAX,X,F,G)

      DOUBLE PRECISION ONE,ZERO
      PARAMETER(ONE=1.0D+0,ZERO=0.0D+0,EPS=0.00001)
      INTEGER NU,NZ,M,K,NMAX,MMAX,IPIV(NMAX),INFO
      DOUBLE PRECISION X(NMAX),F,H(NMAX,NMAX),Y(NMAX),YTP
      DOUBLE PRECISION DNRM2,NPROJE,P(NMAX),PN(NMAX),DDOT,PTBP
      DOUBLE PRECISION TAU(MMAX),WORK(NMAX),G(NMAX),V(MMAX)
      DOUBLE PRECISION A(MMAX,NMAX),AT(NMAX,MMAX),Z(NMAX,MMAX)
      DOUBLE PRECISION C(NMAX,NMAX),D(NMAX,NMAX),B(NMAX,NMAX)
      DOUBLE PRECISION YVELHO(NMAX),YNOVO(NMAX),BP(NMAX),BTP(NMAX)
      DOUBLE PRECISION ALPHA1,ALPHA2
      REAL EPS
      CHARACTER*1 T,n


C     Transposição da matriz A
      DO I=1,NU
         DO J=1,M
         AT(I,J)=A(J,I)
         END DO
      END DO

C     Cálculo da fatoração QR de AT
      CALL DGEQRF(NU,M,AT,NMAX,TAU,WORK,NU,INFO)

C     Gera a matriz Q de AT
      CALL DORGQR(NU,NU,M,AT,NMAX,TAU,WORK,NU,INFO)

C     Extrai as últimas colunas da Q
      DO I=1,NU
         DO J=1,NU
            Z(I,J)=AT(I,J+M)
         END DO
      END DO

      K=0
C     Cálculo do gradiente(G) da função objetivo
      CALL COFG(NU,X,F,G,.TRUE.)

C     Cálculo do produto YVELHO=Z^T*G (Z^T * gradiente)
      NZ=NU-M
      CALL DGEMV('T',NU,NZ,ONE,Z,NMAX,G,1,ZERO,YVELHO,1)

```

```
C     Cálculo da norma do gradiente
      NPROJE=DNRM2(NZ,YVELHO,1)

      DO I=1,NZ
        P(I)=(-1)*YVELHO(I)
      END DO
C     Cálculo do produto PN=Z*P (Z * direçao)
      CALL DGEMV('n',NU,NZ,ONE,Z,NMAX,P,1,ZERO,PN,1)

      K=1
C     Cálculo do X_1
      CALL DAXPY(NU,ONE,PN,1,X,1)

C     Cálculo do gradiente(G) da função objetivo
      CALL COFG(NU,X,F,G,.TRUE.)

C     Cálculo do produto YNOVO=Z^T*G (Z^T * gradiente)
      CALL DGEMV('T',NU,NZ,ONE,Z,NMAX,G,1,ZERO,YNOVO,1)

C     Cálculo da norma do gradiente
      NPROJE=DNRM2(NZ,YNOVO,1)

C     A matriz B inicial é a indentidade
      DO I=1,NZ
         DO J=1,NZ
            IF (I.EQ.J) THEN
               B(I,J)=ONE
            ELSE
               B(I,J)=ZERO
            END IF
         END DO
      END DO

10    IF (NPROJE.GT.EPS) THEN
      DO I=1,NZ
        Y(I)=YNOVO(I)-YVELHO(I)
      END DO

C     Cálculo da aproximação da hessiana (a matriz B)
C     Cálculo do produto BP=B*P
      CALL DGEMV('n',NZ,NZ,ONE,B,NMAX,P,1,ZERO,BP,1)

C     Cálculo de BTP=B^T*P
      CALL DGEMV('T',NZ,NZ,ONE,B,NMAX,P,1,ZERO,BTP,1)

```

```
C     Cálculo do produto PTBP=P^T*BP
      PTBP=DDOT(NZ,P,1,BP,1)

      ALPHA1=(-1)/PTBP
C     Cálculo do produto B=ALPHA1*BP*BTP^T + B
      CALL DGEMM('n','T',NZ,NZ,1,ALPHA1,BP,NMAX,BTP,NMAX,ONE,B,NMAX)

C     Cálculo do produto YTP=Y^T*P
      YTP=DDOT(NZ,Y,1,P,1)

      ALPHA2=1/YTP
C     Cálculo do produto B=ALPHA2*Y*Y^T+B
      CALL DGEMM('n','T',NZ,NZ,1,ALPHA2,Y,NMAX,Y,NMAX,ONE,B,NMAX)

      DO I=1,NU
        P(I)=(-1)*YNOVO(I)
      END DO

C     Resolve  B*p=-YNOVO usando a LU de B
      CALL DGESV(NZ,1,B,NMAX,IPIV,P,NMAX,INFO)

C     Cálculo do produto PN=Z*P (Z * direçao)
      CALL DGEMV('n',NU,NZ,ONE,Z,NMAX,P,1,ZERO,PN,1)

C     Atualiza o valor de X
      CALL DAXPY(NU,ONE,PN,1,X,1)

C     Cálculo do gradiente(G) da função objetivo
      CALL COFG(NU,X,F,G,.TRUE.)

      DO I=1,NZ
        YVELHO(I)=YNOVO(I)
      END DO

C     Cálculo do produto YNOVO=Z^T*G (Z^T * gradiente)
      CALL DGEMV('T',NU,NZ,ONE,Z,NMAX,G,1,ZERO,YNOVO,1)

C     Cálculo da norma do gradiente projetado
      NPROJE=DNRM2(NZ,YNOVO,1)

      K=K+1
      GO TO 10
      END IF
      RETURN
      END
```

Problema	Classificação	Número de variáveis	Número de restrições	Norma do gradiente projetado
BT3	SLR2-AY-5-3	5	3	68.7078542 169.40563 4.11236379 3.73045289 0.02604802 0.00466981027 2.68545372^{-06}
GENHS28	QLR2-AY-10-8	10	8	7.74301427 14.8631173 0.449428842 0.165787813 0.000870170383 9.12697482^{-06}
HS28	SLR2-AY-3-1	3	1	7.46420027 12.5776886 0.802037976 0.430064362 0.00329446168 0.000318949997 4.0781398^{-06}
HS48	SLR2-AY-5-2	5	2	25.0421866 69.5868269 2.21869071 6.24828424 0.281683371 0.151436254 0.00549232846 0.00229458112 9.87730334^{-05} 1.94553009^{-05} 6.84958585^{-07}
HS49	OLR2-AY-5-2	5	2	não converge

Problema	Classificação	Número de variáveis	Número de restrições	Norma do gradiente projetado
HS50	OLR2-AY-5-3	5	3	803.346365 571298765. 1066.43798 833.829033 670.47931 670.414843 199.912055 98.9711569 40.8414637 18.1476596 7.79378438 3.60036056 3.29281288 3.54933556 2.60724214 2.32940641 1.86707456 0.641853967 0.044773316 0.0123868106 0.0010972785 0.000154675836 4.15195515^{-06}
HS51	QLR2-AY-5-3	5	3	6.34580419 11.2753936 3.00002675 2.98305137 0.111233671 0.0191933909 3.58447232^{-05} 1.93923398^{-07}
HS52	QLR2-AY-5-3	5	3	39.9634448 1034.87001 1.89070496 8.82493966 0.12867548 0.127304847 0.00107287871 0.00112321497 9.44026667^{-08}

Observe que o comportamento do método de Broyden é, em alguns casos, oscilatório (observe os problemas HS48, HS50 e HS52). Este fato ocorre porque usamos a matriz identidade como aproximação inicial ($B_0 = I$). Isso poderia ser corrigido, ou pelo

menos minimizado por meio de duas alternativas que são: inicialização com uma matriz mais próxima da Hessiana, ou mesmo a própria Hessiana $B_0 = \nabla^2 f(x_0)$; outra maneira seria introduzir uma técnica de globalização como região de confiança com busca linear. Estas modificações serão feitas com a continuidade de nossos estudos futuramente.

Conclusão

Neste trabalho implementamos métodos para minimização de funções lineares com restrição de igualdade. O principal objetivo do trabalho, a programação e validação da metodologia na plataforma CUTE foi plenamente alcançada.

A experiência deste Trabalho de Conclusão de Curso abre inúmeras possibilidades de pesquisa em otimização. Com acesso a uma variedade de problemas-testes e de um robusto ambiente de programação podemos vislumbrar novos métodos, assim como novas implementações sempre baseadas em estudos teóricos que caracteriza esta importante área da matemática. Uma possível continuidade do trabalho seria a introdução de restrições de desigualdades lineares por meio do método de restrições ativas, ou mesmo restrições não lineares.

Finalmente, por meio de todo o estudo teórico e principalmente computacional que envolveu esse trabalho, estabelecemos condições muito favoráveis para enfrentar um trabalho em nível de pós-graduação.

Referências

[1] GILL, Philip E.; MURRAY, Walter; WRIGHT, Margaret H., *Pratical Otimization*. Academic Press, San Diego, California, 1981.

[2] DENNIS, J. E. Jr; SCHNABEL, Robert B., *Numerical Methods for Unconstrained Optimization and Nonlinear Equations*. SIAM Philadelphia, 1996.

[3] NOCEDAL, Jorge; WRIGHT, Stephen J., *Numerical Optimization*. Springer, 1999.

[4] STRANG, Gilbert, *Linear Algebra and its Applications*. Thomson Learning, 1988.

[5] NEMHAUSER, G. L.; RINNOOY KAN, A. H. G.; TODD, M. J., *Optimization*. Elsevier Science Publishing Company, 1989.

[6] FLETCHER, R., *Pratical Methods of Optimization*. John Wiley & Songs, 1987.

[7] GOULD, N. I. M.; ORBAN, D.; TOINT, Ph. L., *General CUTEr documentation*. Rutherford Appleton Laboratory, UK, CERFACS, France and Facultés Universitaires Notre-Dame de la Paix, Belgium, 2003. ver http://cuter.rl.ac.uk/cuter-www/

[8] PAZ, Juliana A., *O Ambiente CUTE para problemas de Otimização*. Florianópolis, 2004. Monografia (Trabalho de Conclusão de Curso) - Centro de Ciências Físicas e Matemáticas, Universidade Federal de Santa Catarina.